CRISTHIAN ARLINDO MAMANI NINA

January 2025

[12pt]article amsmath amsfonts amssymb geometry a4paper
Informe sobre la Implementación de Métodos de Resolución de Sistemas de Ecuaciones Lineales Cristhian Arlindo January 21, 2025

1 Introducción

Este informe describe la implementación de tres métodos de resolución de sistemas de ecuaciones lineales (SEL) mediante el uso de la librería `Streamlit` para crear una interfaz interactiva. Los métodos implementados son:

- Método de **Sustitución hacia atrás**.
- Método de **Gauss-Jordan**.
- Método de **Cramer**.

El código permite ingresar una matriz de coeficientes A y un vector de términos independientes b para resolver el sistema $A \cdot x = b$, donde x es el vector solución.

2 Descripción de los Métodos

2.1 Sustitución hacia atrás

El método de sustitución hacia atrás es adecuado para sistemas que ya están en forma triangular superior. Se resuelve empezando desde la última ecuación hacia la primera. El código verifica que la matriz sea triangular superior y no tenga ceros en los elementos diagonales principales, lo cual sería un error.

El algoritmo implementado es el siguiente:

1. Comienza con el valor más bajo de $x_n = \frac{b_n}{a_{nn}}$.
2. Luego, para cada ecuación i en el rango $n-1$ a 0, se calcula $x_i = \frac{b_i - \sum a_{ij} x_j}{a_{ii}}$.

2.2 Gauss-Jordan

El método de Gauss-Jordan es una extensión del método de eliminación de Gauss y se utiliza para resolver sistemas de ecuaciones lineales de forma más directa. Este método lleva la matriz aumentada $[A|b]$ a su forma reducida por filas, de modo que los coeficientes de la matriz A sean 1 en la diagonal y 0 en el resto de las posiciones.

El proceso es el siguiente:

1. Dividir cada fila i de la matriz aumentada por el valor de a_{ii}.
2. Luego, para cada $i \neq j$, se realiza una operación de fila $F_j \leftarrow F_j - F_i \cdot a_{ji}$ para hacer ceros en la columna i.

2.3 Método de Cramer

El método de Cramer utiliza determinantes para encontrar las soluciones de un sistema de ecuaciones lineales. Este método se aplica cuando la matriz de coeficientes A tiene un determinante distinto de cero. La solución x_i se obtiene sustituyendo las columnas de la matriz A por el vector b y calculando el determinante de la nueva matriz.

El algoritmo sigue estos pasos:

1. Calcular el determinante de la matriz A.
2. Para cada incógnita x_i, se reemplaza la columna i de A por b y se calcula el determinante de la nueva matriz A_i.
3. La solución de x_i es $x_i = \frac{det(A_i)}{det(A)}$.

3 Implementación en Streamlit

La implementación del código en `Streamlit` permite interactuar con el sistema, ingresando la matriz A y el vector b a través de una interfaz web. El usuario puede seleccionar el tamaño del sistema y el método de resolución de su elección.

El código en `Python` utiliza las librerías `NumPy` y `Streamlit` para realizar las operaciones matemáticas y proporcionar una interfaz gráfica de usuario. El código es el siguiente:

```
import streamlit as st
import numpy as np

def sustitucion(A, b):
    n = len(A)
    x = np.zeros(n)

    # Verificar si la matriz es triangular superior
```

```
    for i in range(n):
        if A[i][i] == 0:
            st.error("El sistema no puede resolverse por sustitución directamente")
            return None

    # Sustitución hacia atrás
    x[n-1] = b[n-1] / A[n-1][n-1]
    for i in range(n-2, -1, -1):
        suma = 0
        for j in range(i+1, n):
            suma += A[i][j] * x[j]
        x[i] = (b[i] - suma) / A[i][i]

    return x

def gauss_jordan(A, b):
    n = len(A)
    Ab = np.concatenate((A, b.reshape(n,1)), axis=1)

    for i in range(n):
        pivot = Ab[i][i]
        if pivot == 0:
            st.error("El sistema no puede resolverse por Gauss-Jordan")
            return None

        Ab[i] = Ab[i] / pivot

        for j in range(n):
            if i != j:
                Ab[j] = Ab[j] - Ab[i] * Ab[j][i]

    return Ab[:, -1]

def cramer(A, b):
    n = len(A)
    det_A = np.linalg.det(A)

    if det_A == 0:
        st.error("El sistema no puede resolverse por Cramer (determinante = 0)")
        return None

    x = np.zeros(n)

    for i in range(n):
        Ai = A.copy()
        Ai[:, i] = b
```

```
        x[i] = np.linalg.det(Ai) / det_A

    return x

def main():
    st.title("Sistema de Ecuaciones - Método SEL")
    st.write("Implementación de Sustitución, Gauss-Jordan y Cramer")

    n = st.number_input("Tamaño del sistema (nxn)", min_value=2, max_value=5, value=3)

    st.write("Ingrese los coeficientes de la matriz A:")
    A = np.zeros((n,n))
    for i in range(n):
        cols = st.columns(n)
        for j in range(n):
            A[i,j] = cols[j].number_input(f"A[{i+1},{j+1}]", value=0.0)

    st.write("Ingrese los términos independientes (vector b):")
    b = np.zeros(n)
    cols = st.columns(n)
    for i in range(n):
        b[i] = cols[i].number_input(f"b[{i+1}]", value=0.0)

    metodo = st.radio(
        "Seleccione el método de solución:",
        ["Sustitución", "Gauss-Jordan", "Cramer"]
    )

    if st.button("Resolver"):
        if metodo == "Sustitución":
            x = sustitucion(A, b)
        elif metodo == "Gauss-Jordan":
            x = gauss_jordan(A, b)
        else:
            x = cramer(A, b)

        if x is not None:
            st.write("Solución encontrada:")
            for i in range(n):
                st.write(f"x{i+1} = {x[i]:.4f}")

if __name__ == "__main__":
    main()
```

4 Enlaces Relevantes

El código completo de este proyecto está disponible en GitHub. Puedes acceder a él a través del siguiente enlace:

https://github.com/cristhian-arlindo16/pip/commit/1d760713c13b0e821dac4d4896a9cee45be64687

Además, puedes interactuar con la aplicación desplegada en Streamlit en el siguiente enlace:

https://d8xmhv3vdkv82diwhe8vve.streamlit.app/